Roll No. Is : DS5B-2118

Roll No. Is : DS5B-2118

Roll No. Is : DS5B-2118

Roll No. Is : DS5B-2118

Roll No. Is : DS5B-2118

Roll No. Is : DS5B-2118

Connecting To The MongoDB Compass"

In [ ]:

```
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publ
ic/simple/
Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
     |████████████████████████████████| 281.4 MB 29 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
     |████████████████████████████████| 198 kB 42.5 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl size=281853642
sha256=8b46831582fe33020b51646e24a86fc72a74a84b94240776aece0e2d97207751
  Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd8017084dce4e93e84e92efd1e1d53
34db05f2e83bcef74f
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.3 pyspark-3.2.1
```

In [ ]:

```
from pyspark.sql import SparkSession
from pyspark.ml import Pipeline
from pyspark.ml.feature import VectorAssembler, StringIndexer, OneHotEncoder
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator

session = SparkSession.builder.appName("HR_Dataset").getOrCreate()
data = session.read.csv("HR comma.csv", header = True, inferSchema = True)
```

In [ ]:

```
data.show(10)
```

```
+------------------+---------------+--------------+--------------------+----------------
--+-------------+----+---------------------+-----+------+
|satisfaction_level|last_evaluation|number_project|average_montly_hours|time_spend_compa
ny|Work_accident|left|promotion_last_5years|sales|salary|
+------------------+---------------+--------------+--------------------+----------------
--+-------------+----+---------------------+-----+------+
|              0.38|           0.53|             2|                 157|
3|            0|   1|                    0|sales|   low|
|               0.8|           0.86|             5|                 262|
6|            0|   1|                    0|sales|medium|
|              0.11|           0.88|             7|                 272|
4|            0|   1|                    0|sales|medium|
|              0.72|           0.87|             5|                 223|
5|            0|   1|                    0|sales|   low|
|              0.37|           0.52|             2|                 159|
3|            0|   1|                    0|sales|   low|
|              0.41|            0.5|             2|                 153|
3|            0|   1|                    0|sales|   low|
|               0.1|           0.77|             6|                 247|
4|            0|   1|                    0|sales|   low|
|              0.92|           0.85|             5|                 259|
5|            0|   1|                    0|sales|   low|
|              0.89|            1.0|             5|                 224|
5|            0|   1|                    0|sales|   low|
|              0.42|           0.53|             2|                 142|
3|            0|   1|                    0|sales|   low|
+------------------+---------------+--------------+--------------------+----------------
--+-------------+----+---------------------+-----+------+
only showing top 10 rows
```

In [ ]:

```
data.columns
```

```
['satisfaction_level',
 'last_evaluation',
 'number_project',
 'average_montly_hours',
 'time_spend_company',
 'Work_accident',
 'left',
 'promotion_last_5years',
 'sales',
 'salary']
```

In [ ]:

```
str_idxer = StringIndexer(inputCols = ['sales','salary'], outputCols = ["newsales", "new
```

In [ ]:

```
one_hot_encoding = OneHotEncoder(inputCols = ["newsales","newsalary"], outputCols = ["ne
```

In [ ]:

```
vec_ass = VectorAssembler(inputCols = ['satisfaction_level','last_evaluation','number_pr
```

In [ ]:

```
lr = LogisticRegression(featuresCol= "all_features", labelCol = "left")
```

In [ ]:

```
mypipeline = Pipeline(stages = [str_idxer, one_hot_encoding, vec_ass, lr])
```

In [ ]:

```
training, test = data.randomSplit([0.71, 0.29])
```

In [ ]:

```
lr_model = mypipeline.fit(training)
```

In [ ]:

```
result = lr_model.transform(test)
```

In [ ]:

```
result.show(4, truncate = False)
```

```
+------------------+---------------+--------------+--------------------+----------------
--+-------------+----+---------------------+-----------+------+--------+---------+------
---------+----------------+--------------------------------------------------------+----
------------------------------------+----------------------------------------+----------
+
|satisfaction_level|last_evaluation|number_project|average_montly_hours|time_spend_compa
ny|Work_accident|left|promotion_last_5years|sales      |salary|newsales|newsalary|newsal
es_onehot|newsalary_onehot|all_features                                            |rawP
rediction                           |probability                             |prediction
|
+------------------+---------------+--------------+--------------------+----------------
--+-------------+----+---------------------+-----------+------+--------+---------+------
---------+----------------+--------------------------------------------------------+----
------------------------------------+----------------------------------------+----------
+
|0.09              |0.62           |6             |294                 |4
```

```
  |0             |1   |0                     |accounting |low   |8.0     |0.0      |(9,
[8],[1.0])  |(2,[0],[1.0])   |(18,[0,1,2,3,4,15,16],[0.09,0.62,6.0,294.0,4.0,1.0,1.0])|
[-0.906113009228541,0.906113009228541]  |[0.28779589386338533,0.7122041061366147]|1.0
    |
|0.09              |0.62            |6              |294                  |4
  |0             |1   |0                     |accounting |low   |8.0     |0.0      |(9,
[8],[1.0])  |(2,[0],[1.0])   |(18,[0,1,2,3,4,15,16],[0.09,0.62,6.0,294.0,4.0,1.0,1.0])|
[-0.906113009228541,0.906113009228541]  |[0.28779589386338533,0.7122041061366147]|1.0
    |
|0.09              |0.77            |5              |275                  |4
  |0             |1   |0                     |product_mng|medium|4.0     |1.0      |(9,
[4],[1.0])  |(2,[1],[1.0])   |(18,[0,1,2,3,4,11,17],[0.09,0.77,5.0,275.0,4.0,1.0,1.0])|
[-0.6706336468375675,0.6706336468375675]|[0.3383549711925885,0.6616450288074115] |1.0
    |
|0.09              |0.77            |6              |244                  |4
  |0             |1   |0                     |product_mng|low   |4.0     |0.0      |(9,
[4],[1.0])  |(2,[0],[1.0])   |(18,[0,1,2,3,4,11,16],[0.09,0.77,6.0,244.0,4.0,1.0,1.0])|
[-0.7998272517122431,0.7998272517122431]|[0.31006247261884357,0.6899375273811564]|1.0
    |
+------------------+---------------+--------------+---------------------+----------------
--+-------------+----+---------------------+-----------+------+--------+---------+------
---------+----------------+---------------------------------------------------------+----
------------------------------------+----------------------------------------+----------
+
only showing top 4 rows
```

In [ ]:
```python
eval = BinaryClassificationEvaluator(rawPredictionCol="rawPrediction", labelCol = "left"
```

In [ ]:
```python
eval.evaluate(result)
```

```
0.8297104126919155
```

In [1]:
```python
import pymongo  # DS5B 2112
client = pymongo.MongoClient("mongodb://localhost:27017/")
client
```

Out[1]:
```
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)
```

List Of Databases

In [2]:
```python
client.list_database_names()  # DS5B 2112
```

Out[2]:
```
['India', 'admin', 'config', 'local']
```

Create DataBase

In [3]:
```python
mydb = client["Hospital_Data"]  # the DB will not show in list untill unless data is ins
```

Creating Collection inside DataBase

In [4]:
```python
collection1 = mydb["patient_info"]
```

Inserting a Single record

In [7]:
```python
record = {"Patient1":{"Name":"Meghanshu kumrawat",
              "Age":25,
              "city":"Indore",
              "contact no":[7047436645,8982161294],
              "patient gaurdian":{"G_name":"Anil jain","G_age":34,"G_realtion":"father"},
              "dieseas":"heart attack",
              "condition":"critical",
             "report":{"blood report":"document path", "urin report":"document path", "BP
         }}
 collection1.insert_one(record)
```

Out[7]:

```
<pymongo.results.InsertOneResult at 0x17bf9236fb0>
```

Deleting A Record

In [6]:

```
collection1.delete_one({"Patient1.Name":"Meghanshu kumrawat"})
```

Out[6]:

```
<pymongo.results.DeleteResult at 0x17bf9236fe0>
```

Updating A Record

In [8]:

```
collection1.update_one({"Patient1.Age":25}, {"$set":{"Patient1.Age":60}})
```

Out[8]:

```
<pymongo.results.UpdateResult at 0x17bf9237e80>
```

In [15]:

```
for i in collection1.find():
    print(i)
```

```
{'_id': ObjectId('6292f3a9d3b7bf218cd58df0'), 'Patient1': {'Name': 'Himanshu jain', 'Ag
e': 30, 'city': 'Indore', 'contact no': [7047436645, 8982161294], 'patient gaurdian':
{'G_name': 'Anil jain', 'G_age': 34, 'G_realtion': 'father'}, 'dieseas': 'heart attack',
'condition': 'critical'}}
```

Search a record

In [9]:

```
query1 = {"Patient1.Name":"Meghanshu kumrawat"}

results = collection1.find(query1)
for data in results:
    print(data)
```

```
{'_id': ObjectId('62930dd5202b6c4e57b5c287'), 'Patient1': {'Name': 'Meghanshu kumrawat',
'Age': 60, 'city': 'Indore', 'contact no': [7047436645, 8982161294], 'patient gaurdian':
{'G_name': 'Anil jain', 'G_age': 34, 'G_realtion': 'father'}, 'dieseas': 'heart attack',
'condition': 'critical', 'report': {'blood report': 'document path', 'urin report': 'doc
ument path', 'BP report': 'document path'}}}
```

In [ ]: